High Performance Computing Facilities for the Next Millennium

The NERSC Effective System Performance Test

SC 99 Tutorial
November 14, 1999

Adrian Wong, Lenny Oliker, Bill Kramer,
Teresa Kaltz, David Bailey

kramer@nersc.gov
(510) 486-7577

Discussion Goals

- Why is a test necessary
- NERSC's Effective System Performance (ESP) test framework
- Initial reference implementation
- Results to date

Three Steps to getting effective systems

1) Determine primary indicators (metrics) that are most meaningful
   - Set target goals
   - Goals need to be integrated - service metrics as well as system metrics

2) Manage systems/methods and measure performance towards goals

3) Establish ways to predict performance of new systems/methods
   - Before purchase or during evolution of system

NERSC is now focusing on the third step

- NERSC established effective goals and managed towards them for systems and services since 1996
  - Goals established yearly in a formal process with users and stake holders
  - All goals have primary indicator metrics
    - Many additional metrics and data used as well
  - Goals cover systems and services
- NERSC is doing well in meeting and/or exceeding metrics.
  - Metrics change (and typically targets get higher) as they are consistently met
- NERSC-3 experience indicates we can and should develop a priority measures to predict how effectively "ultra-scale" systems can be.

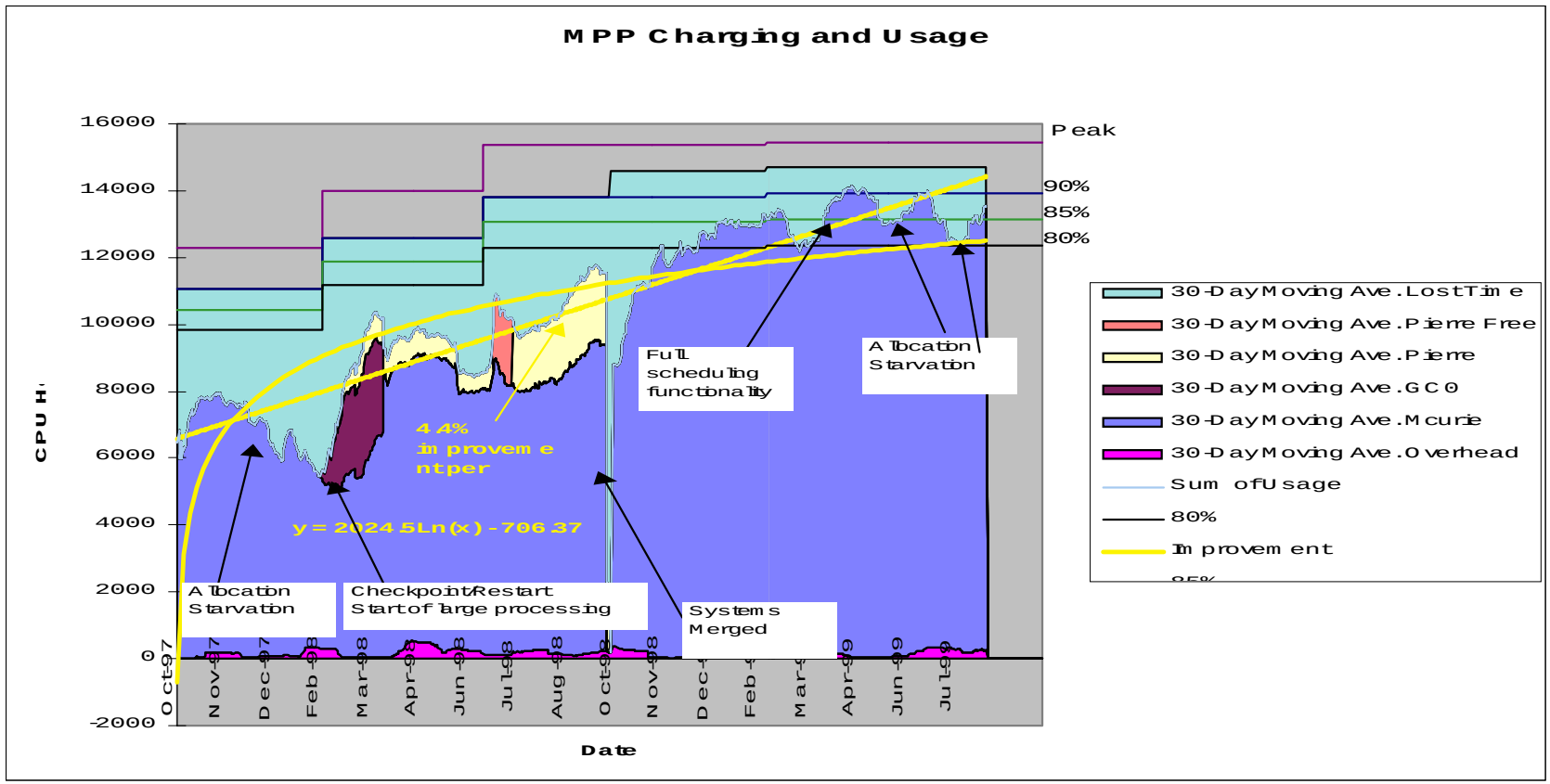
Traditional Benchmarking Approach

- Traditional Benchmarks consist of
  - Dedicated System Tests
    - Some “de facto” standard simple CPU performance tests
      - e.g. LINPACK, SPEC, NPB
    - Maybe a small number of pseudo-applications
      - Mostly limited problem sets for reproducibility
      - Maybe some application kernals
    - An I/O test
      - Disk, maybe internal and external network
  - Sometimes includes a structured throughput workload test

What the Traditional Approach Does Not Measure

- Integrated System function
  - Jobs with varying degrees of CPU, Memory and I/O requirements
- Random mixes of jobs
- System Administration/Resource Management
- System behavior
- Useability
- Slowdown vs Utilization.

Evolution of T3E Utilization



Impact of Increased Effectiveness

- Why spend the effort to increase efficiency
- Increasing efficiency from 80% to 90%
  - 644 PE running at 90% is the equivalent of 725 PE running at 80%
  - 81 PE are needed to make up the difference
  - PE costs ~$50,000 list, $25,000 discounted
- Increased effectiveness is equal to $2M in increased hardware
- Over 18 months, effectiveness increase from ~55% to ~90% - a value of $10.25M
  - In essence this almost the same as Moore's Law improvements in price performance.

Why Systems Improve

- Additional system management tools are made available by the vendor and/or sites.
- Existing system management tools become more effective and robust.
- The user workload stabilizes.
- Users learn how to adjust their jobs to best utilize the system.
- System managers learn how to schedule the user workload and to best use the available system management tools.
- Compilers, I/O and other system software facilities improve.

Integrated Resource Management

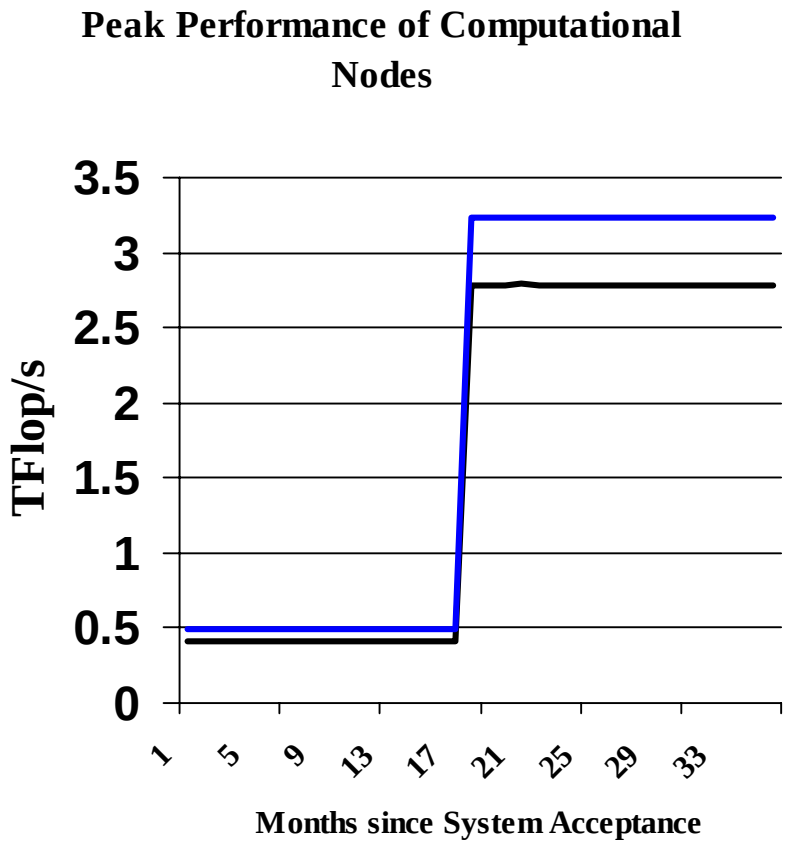
- CPU scheduling could be considered solved
  - Successful implementations
    - Gang Scheduling
    - Priority Scheduling
    - Share Scheduling
  - CPU is no longer the only expensive item in a system
  - Remaining problem is one of firm requirements, not research
- Memory Scheduling a critical factor
  - Memory hierarchy is very complex and will increase in complexity
  - Model of memory scheduling is still essential the simple SMP
  - Not even syntax to describe requirements of a job
  - Memory scheduling is not integrated with CPU scheduling
- Communication Interconnect
- File Systems
- Network
  - Quality of Service integration

Peak Measures Do NOT Indicate a System is Effective for Science

- Peak Operations/sec is a very misleading measure of system performance
  - Says nothing about how much performance can be applied to scientific codes
- Percent of Peak Performance achieved varies widely
  - T3E as an example
    - 644 processors at 900 Mflop/s PE = 580 Gflop/s Peak
    - NPB measured 29.6 Gflop/s for the system
      - ~46 MFlop/s/PE
      - 5.1% of peak
    - Studies of major NERSC applications indicate system is about 67 Gflop/s
      - ~104 MFlop/s/PE
      - 11.6% of peak
    - Gordon Bell prize winning code LSMS was 256 Gflop/s
      - ~398 MFlop/s/PE
      - 44.1 % of peak

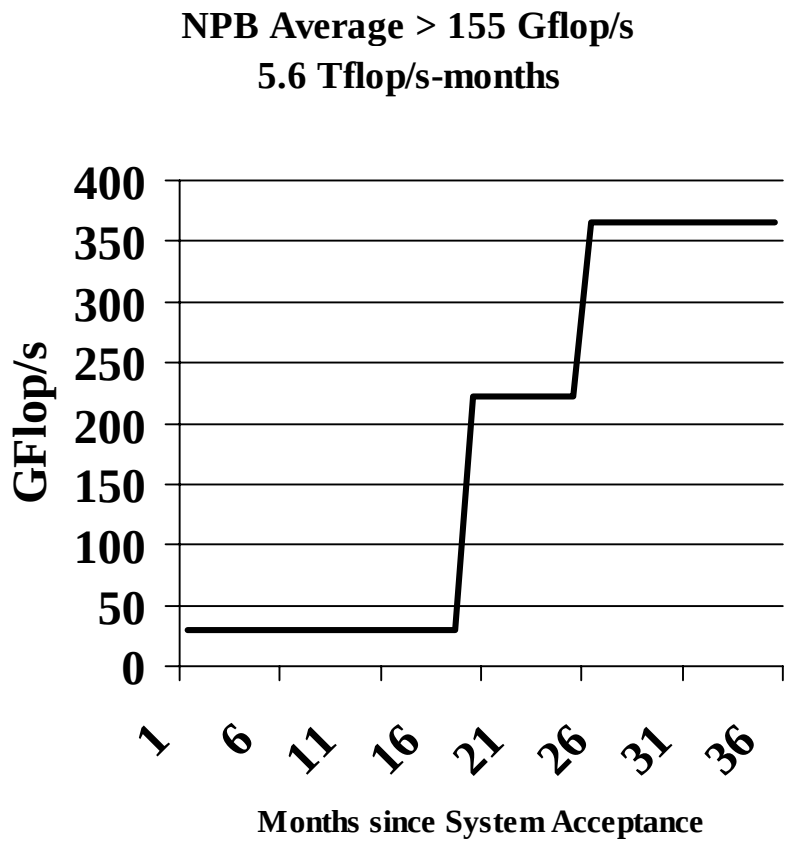
Peak System Performance Measure

- Peak performance of only the number of nodes dedicated to computation
  - 256 in Phase 1
  - 128 in Phase 2
- Vendor projections are 21.6 Gflop/s per node as a minimum in phase 2



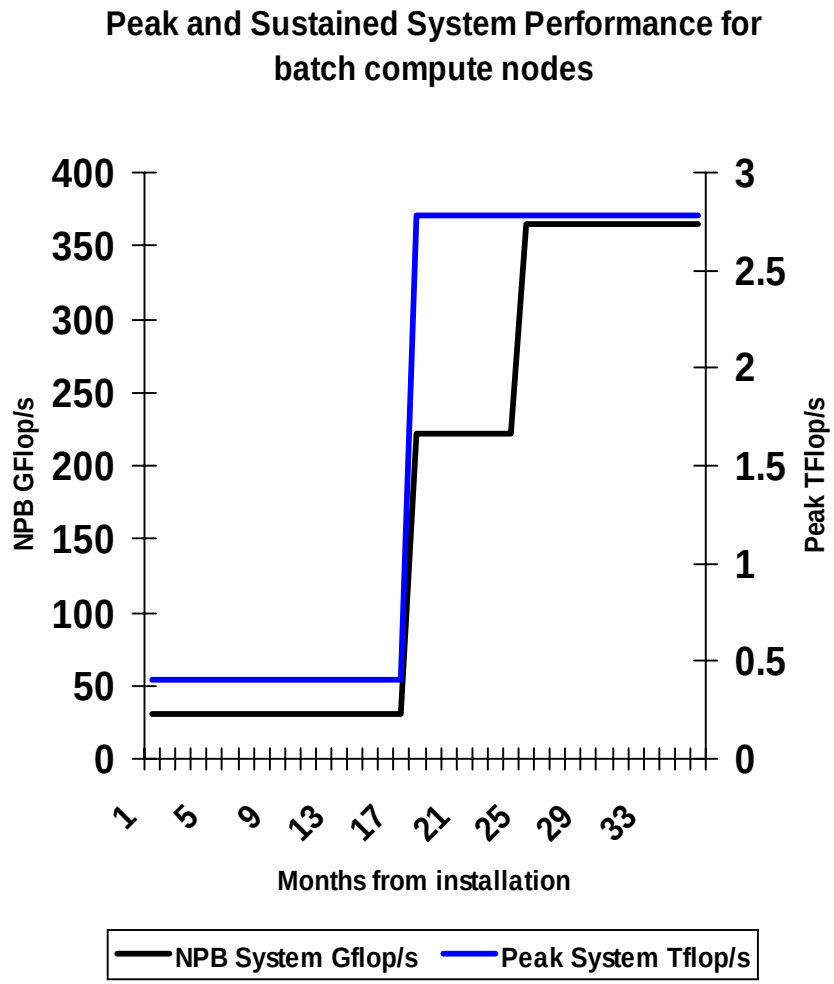
Sustained System Performance Measure

- NPBs are a tough but honest measure for vendors
  - NPBs indicate T3E is a 30 GFlop/s system yet Gordon Bell prize code runs at >250 GFlop/s
  - NPBs typically indicate the lower level of what a good code should get
- Vendor projections are <130 Gflop/s but they committed to meet this measure
  - by faster CPUs,
  - earlier delivery of Phase 2a/b
  - more CPUs

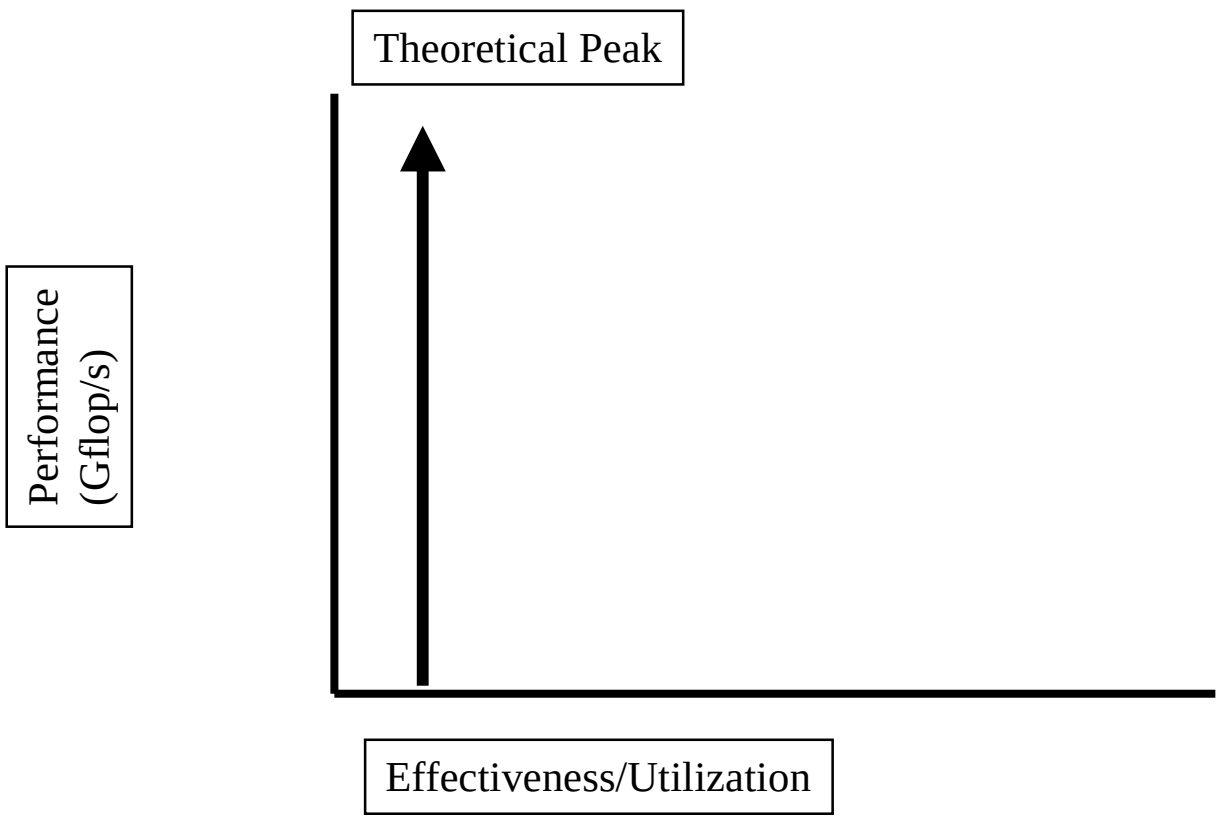


Sustained System Performance a good incentive

- Estimates the amount scientific computation that can really be delivered over time for a system of a constant effectiveness
  - Peak performance is misleading
  - Indicate the lower level of what a good code should get
- Motivated earlier delivery of technology
  - but only when it can be measured and is usable by scientific codes



Benchmark Results



Need for a New Metric -Not just Gflop/s but Effectiveness

- Maximum CPU Utilization (ala GAO) is the morale equivalent to Peak Performance
  - Provides no insight into how well a system is run or how effective it is.
- The ultimate measure is how much science is accomplished with these systems - but no one knows how to measure that
- Why do we use peak CPU?
  - History
    - Useful when the CPU was the only major expense
    - Easy to maximize with traditional CRAY systems
  - So simple, and so uninformative
  - It is hard to measure the really important things
- Why it persists
  - We never defined anything better

New Metric

- We offset misleading Peak Performance with real benchmarks
  - How much effort goes into benchmarking
  - We do nothing to offset CPU maximum utilization
- We should be able to offset misleading CPU utilization with other measures
  - Expected use
    - Determine function of the system and then measure how well it meets function
    - Given function, determine how much CPU time is maximum
  - Usability
    - Throughput measures
  - Total system usage (CPU/Memory/Disk)
  - Peak vs overall

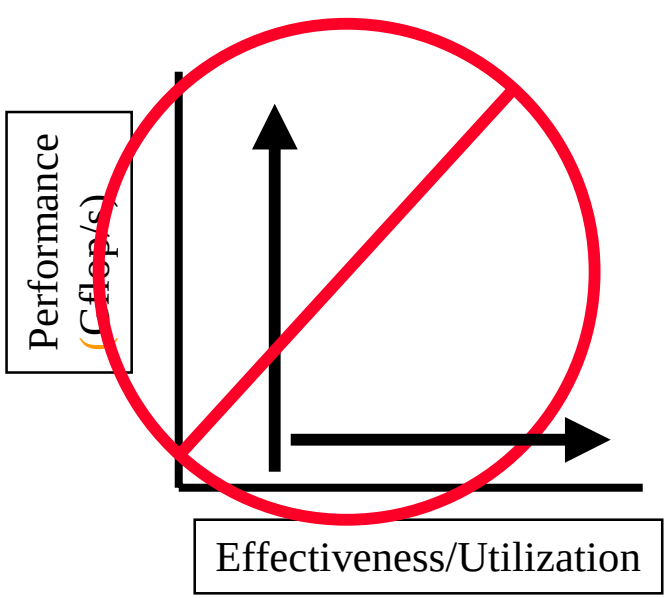
System Measures

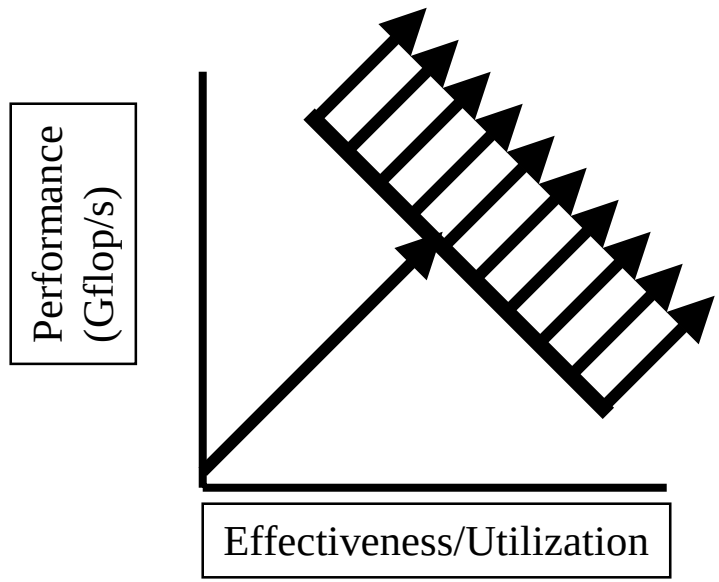
Combining Measures

- **Performance**
  - **How much scientific work can be done for a given quantum of CPU time**
    - **Peak Flop/s**
    - **Measured Flop/s**
- **Effectiveness**
  - **How many quanta of CPU time can be made available to scientific programs over a fixed time period**
    - **CPU time billed vs. theoretical time**
      - **e.g. GAO Report from 1997**

Combining Measures

- **Needed an innovative test to set goals for improving system effectiveness**





The Effective System Performance (ESP) Framework

- The Concept is to simulate “A day in the life of a MPP”
- The Effective System Performance (ESP) test is being developed now
- The goal is to have a measure the predicts effectiveness rather than just measures levels of utilization after the fact
  - Before systems are purchased
  - Evaluate system designs before design and implementation
  - Evaluate system changes before implementation

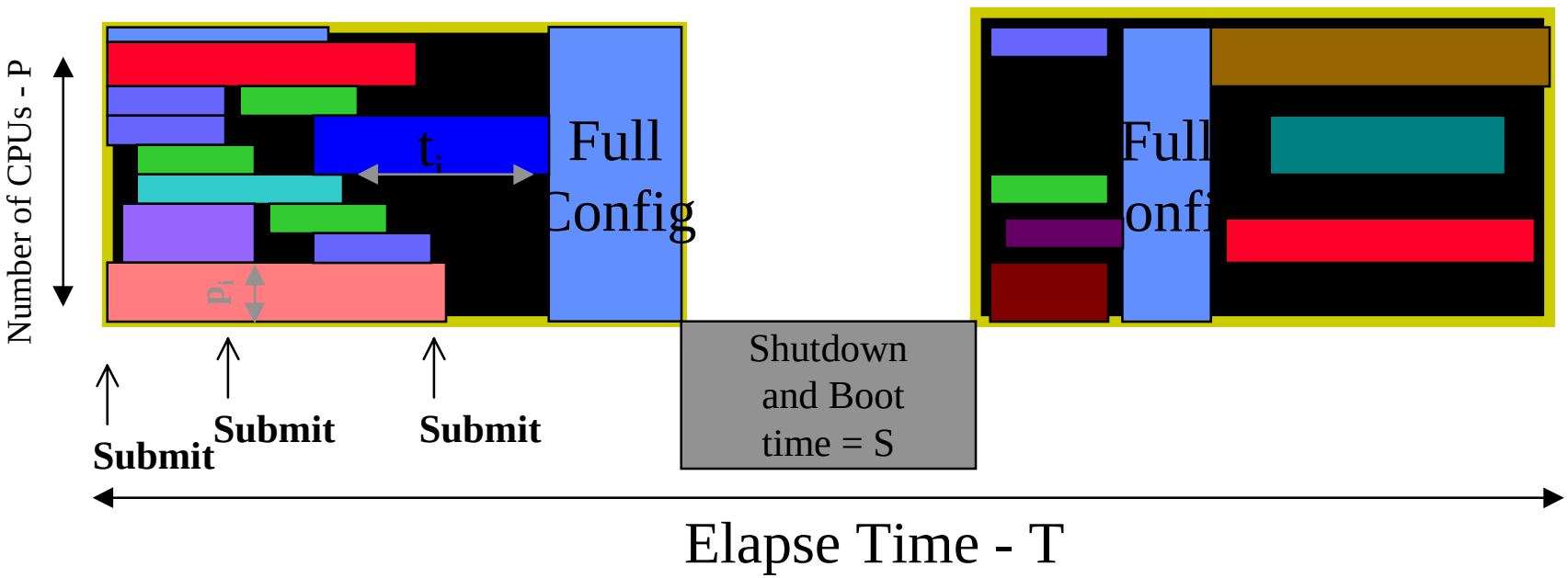
The Effective System Performance (ESP) Measure

- Designed to evaluate systems for overall effectiveness independent of single processor performance
- Looks at overall system
  - Hardware (CPU, memory, disk), software, system management functionality
- Came from trying to prioritize all the possible approaches of improving IBM system software for NERSC-3
  - Not clear what was the best tradeoff until experiment can be performed
  - Composite tests are more amenable to vendors
- Being designed as a general framework as well as a specific test for NERSC-3/4-5

ESP Goals

- Determine how well an existing system supports a particular scientific workload
- Assess systems for that workload before purchase
- Provide quantitative information regarding system enhancements
- Compare different systems on a single workload or discipline
- Compare system-level performance on workloads derived from different disciplines
- Compare different systems for different workloads

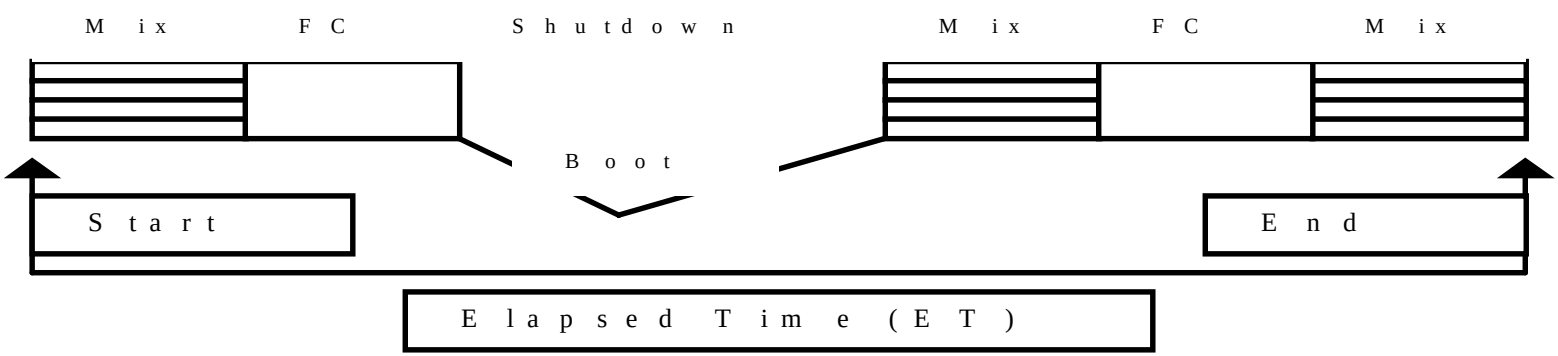
ESP Framework



$$\text{Effectiveness} = (\sum_{i=1,N} p_i * t_i)/[P*(S+T)]$$

ESP

- Test uses a MIX of NERSC test codes, that run in a random order, testing standard system scheduling. There are also Full Configuration codes, I/O tests and typical System Administration activities.
  - Baseline set on SP after Phase 1 acceptance
- Expected yearly improvement, independent of hardware and compiler optimization improvements
- The test measures both how much and how often the system can do scientific work



Results to Date

As of October 1, 1999

NERSC ESP Development Steps

- Determine the right job mix
  - Set up simulation to try different ideas
    - Job size, length, I/O, memory interactions
    - Deterministic vs. random
      - Whether results should be completely reproducible
    - Determine whether to include interactive
- Determine how to submit work to the system
  - All at first, chained, block submissions, random trickle
  - How to include shutdown and boot
- Create scripts and applications and baseline
- Do validation runs - start on T3E
  - Validate against accounting data
- Run on SP to set baseline
- Run ESP with each system improvement
- Tune test with experience

Simulation Studies

- Predicts the expected performance and sensitivity of workloads
- Simulates different workload mixes
- Establishes the best case scheduling estimates
  - First Come First Serve, Best Fit First, Backfill, Checkpoint/restart, gang scheduling
- Helps determine the impact of system functions
- Helps determine the impact of scheduling methods
- Helps determine submission tradeoffs
- Estimate results until sufficient systems can be tested

Simulation Results

	Efficiency				
Schedule Strategy	All at t=0	Dribble	Block	Chain20	Chain30
Next fit first		1.27	1.29	1.30	1.39	1.29
Min processor first	1.25	1.28	1.29	1.40	1.27	
Max processor first	1.21	1.28	1.31	1.41	1.26	
Min time first		1.19	1.31	1.34	1.39	1.30
Max time first		1.06	1.18	1.22	1.39	1.19

Efficiency = simulated time / optimum time, averaged over 100 runs

Based on these results, we prefer multiple blocks of submissions.

ESP Simulation Results

- Simulation Results using best fit
  - Theoretical Best Time 4.04 Hours
  - No Back fill, No C/R 8.28
  - Back fill, No C/R 7.35
  - Back fill with C/R 4.81
  - Gang Scheduling 4.72

    (2 time over subscription, time slice = 1000 sec)
- System Costs not accounted
  - I/O overhead, processors fragmentation, swapping

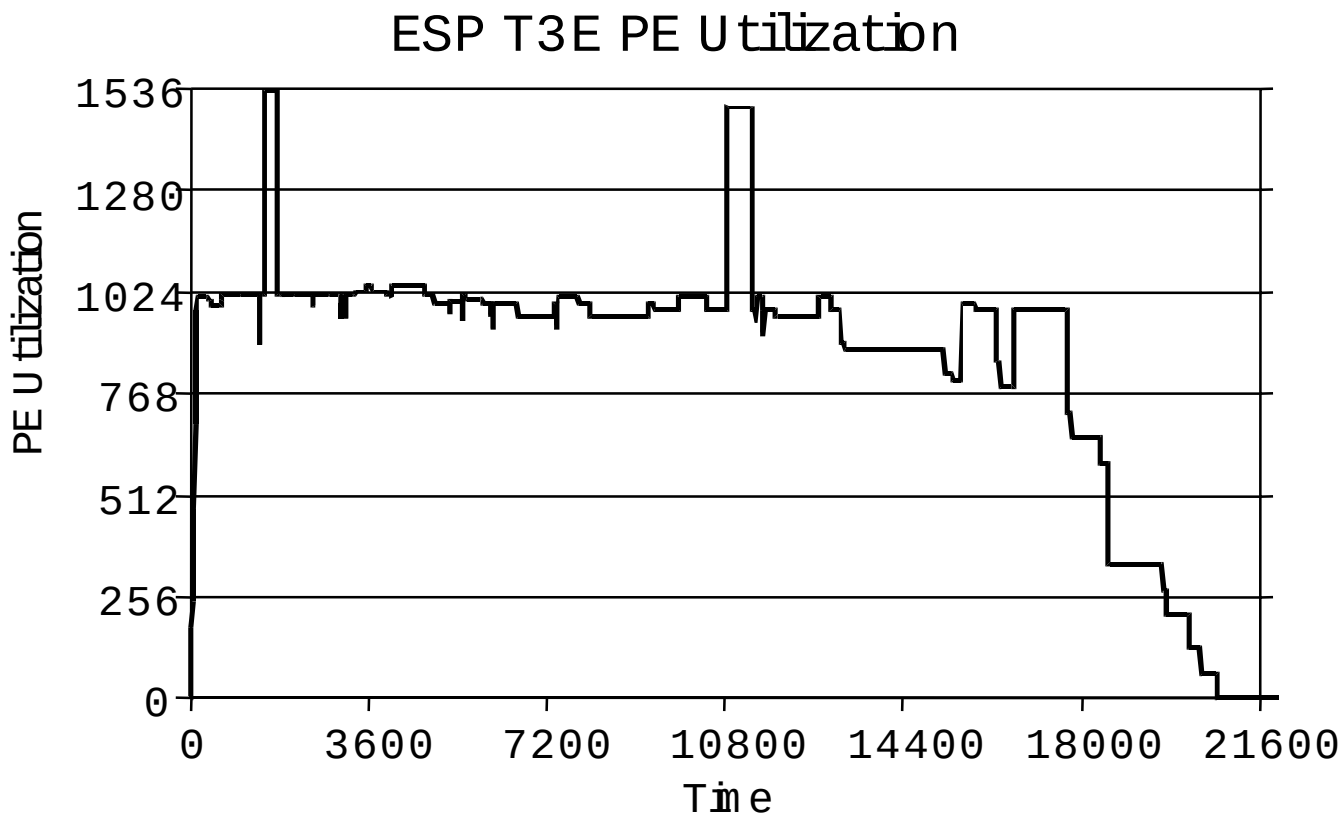
NERSC Workload

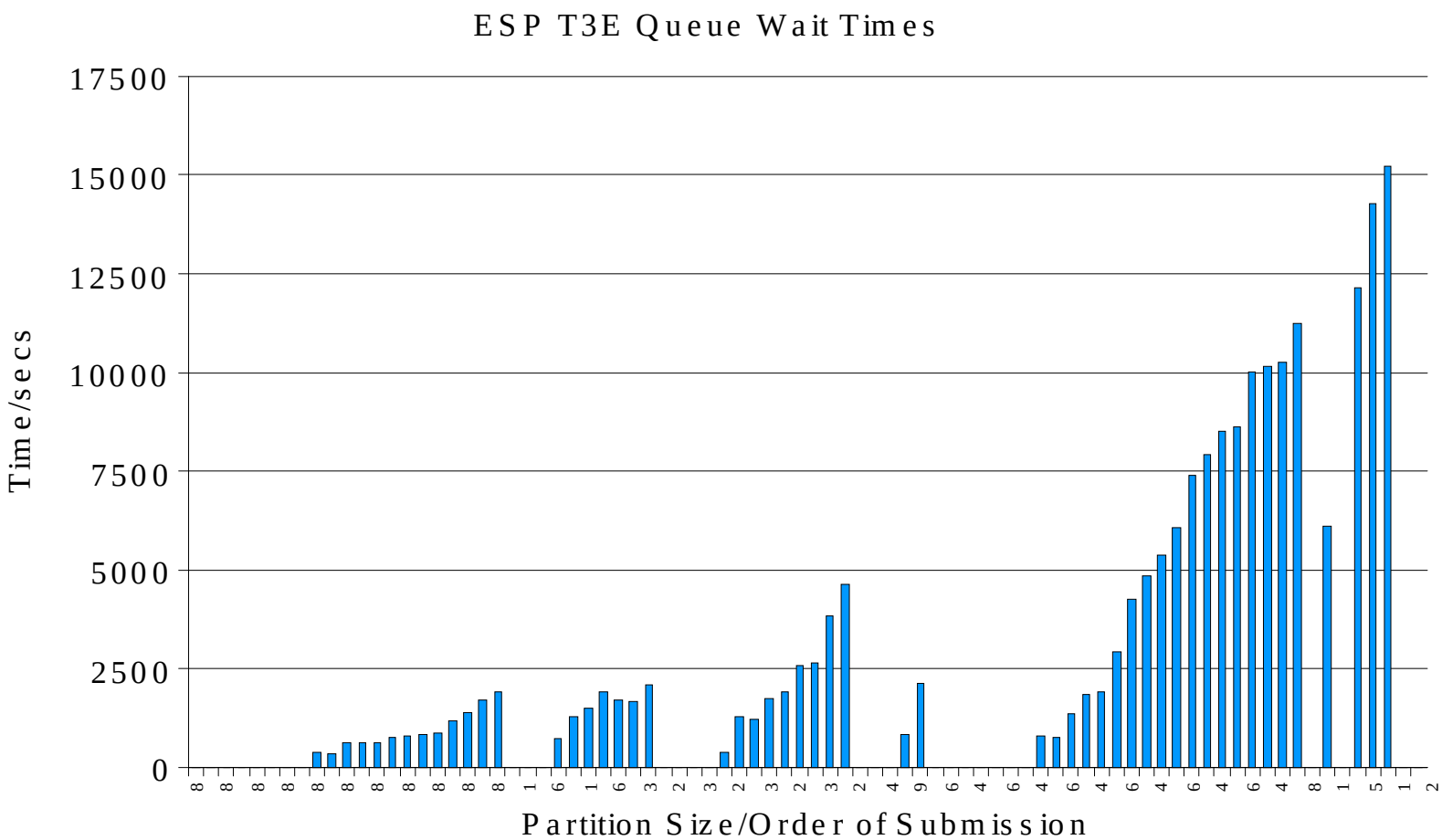
Code Name	Discipline	Number of CPUs	Individual Instances	Percent T3E time	of total
gfft	Large FFT	512	2	30.5	0.42
md	Chemistry	8	4	1208.0	0.52
md		24	3	602.7	0.58
nqclarge	Chemistry	8	2	8788.0	1.89
nqclarge		16	5	5879.6	6.32
paratec	Material science	256	1	746.0	2.57
qcdsmall	Nuclear physics	128	1	1155.0	1.99
qcdsmall		256	1	591.0	2.03
scf	Chemistry	32	7	3461.1	10.42
scf		64	10	1751.9	15.08
scfdirect	Chemistry	64	7	5768.6	34.75
scfdirect		81	2	4578.0	9.97
superlu	Linear algebra	8	15	288.3	0.47
tlbebig	Fusion	16	2	2684.5	1.16
tlbebig		32	6	1358.3	3.51
tlbebig		49	5	912.0	3.00
tlbebig		64	8	685.8	4.72
tlbebig		128	1	350.0	0.60

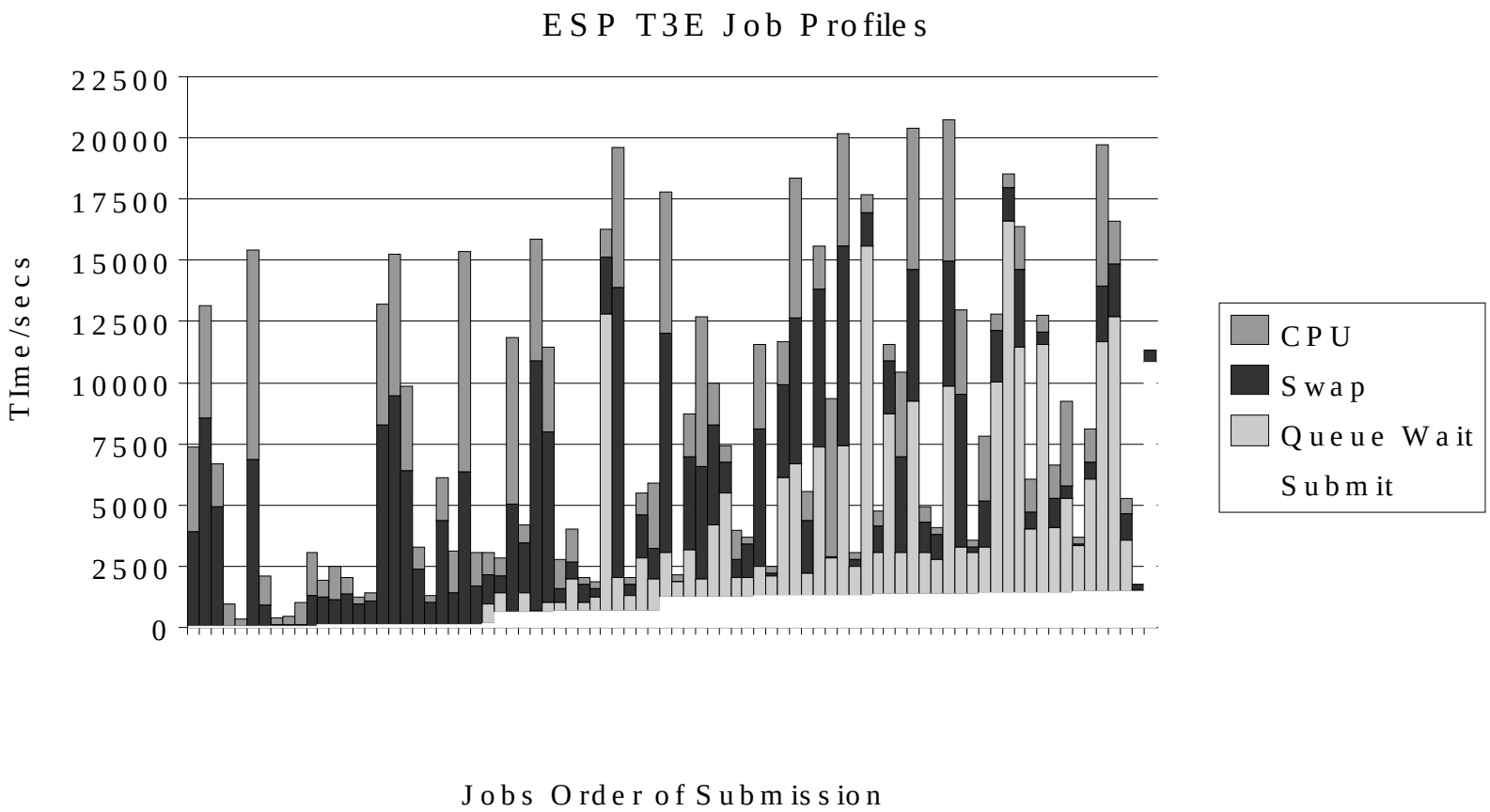
Table 1: The Mix Suite

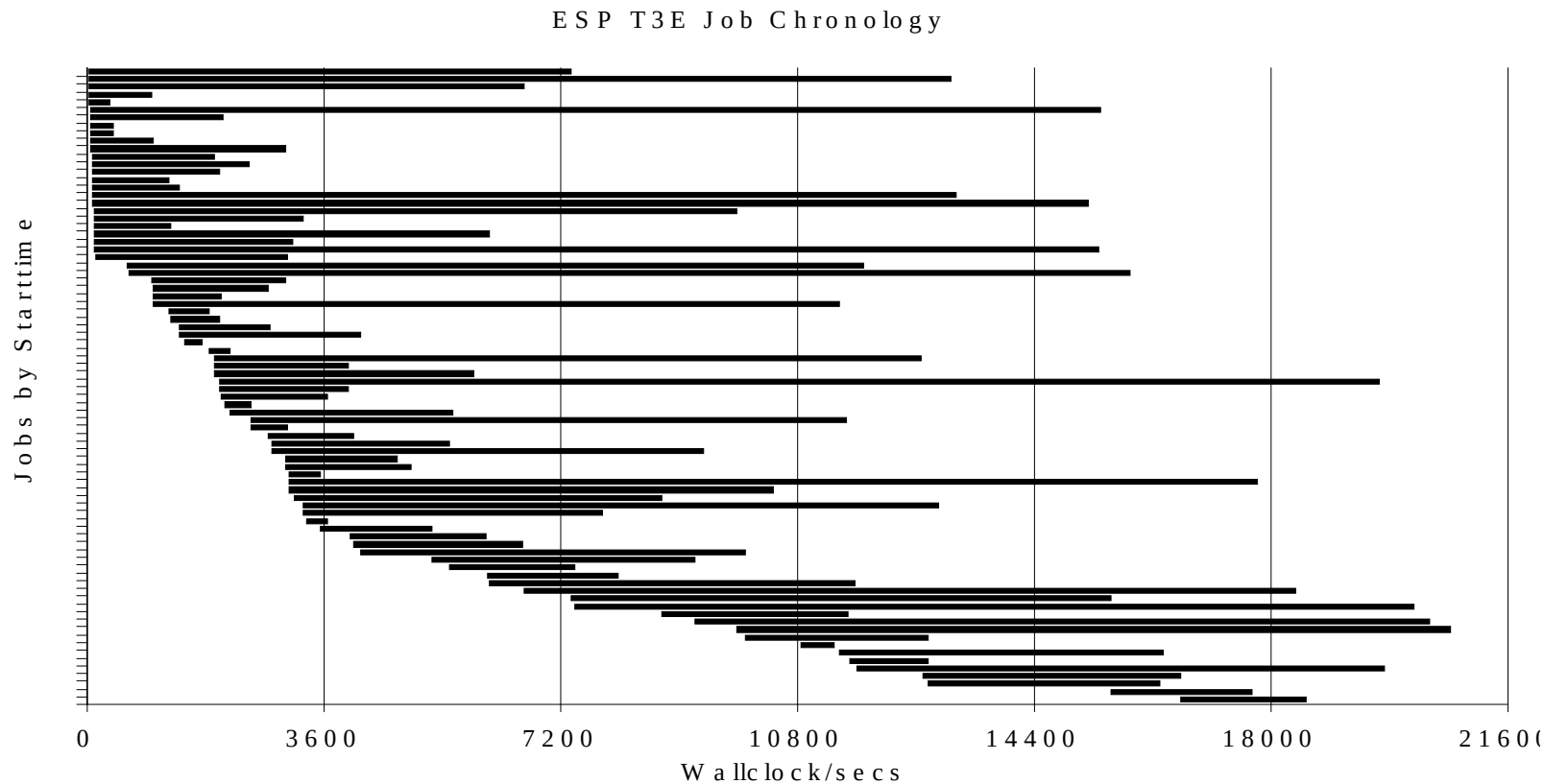
NERSC-3 ESP

- 80 applications, 2 FC jobs
- Job submission - randomly selected
  - Time 0 minutes - jobs submitted to be twice the number of CPUs as system has
  - Time 10 minutes - more jobs submitted until at least the number of CPUs the system has
  - Time 20 minutes - all other jobs submitted
  - Time 24 minutes - first FC job submitted - to run ASAP
  - Time 180 minutes - second FC job submitted


ESP T3E PE Utilization
PE Utilization
1536
1280
1024
768
512
256
0
0
3600
7200
10800
14400
18000
21600
Time


ESP T3E Queue Wait Times
Time/secs
17500
15000
12500
10000
7500
5000
2500
0
Partition Size/Order of Submission


ESP T3E Job Profiles
Time/secs
22500
20000
17500
15000
12500
10000
7500
5000
2500
0
Jobs Order of Submission
CPU
Swap
Queue Wait
Submit


ESP T3E Job Chronology
Jobs by Starttime
0
3600
7200
10800
14400
18000
21600
Wallclock/secs

T3E Results

- First run with original simple job scheduling
  - Two queues
    - One for all none FC jobs, One for two FC jobs
  - First Fit scheduling
  - Allowed gang scheduling (2 time over allocated)

- Simple scheduling similar to the scheduling (without gang scheduling) for early part of T3E profile
  - Actual measured utilization was 63.9% after the first 30 days
    - included approximately 8 hours of down time a week - 5%

T3E Results

- **ESP Test**
  - **Number of processors 512**
  - **Sum of run times 7,437,476 seconds for all jobs**
  - **Elapsed time - 20,739 second without system shutdown**
  - **Average system shutdown recorded - 2,100 seconds**

- **Effectiveness compares will with actual utilization data**
  - **66.8% Effectiveness with system shutdown**
  - **74.0% Effectiveness without system shutdown**

T3E Results

- Workload ran correctly and in a reasonable time!
- BFF pushed large jobs to the end, small jobs first
  - Job starvation at the end of test
  - FC jobs ran appropriately (when they were supposed to)
- Details of test raise more questions
  - Gang scheduling may harm effectiveness (at least for single processor MPPs)
  - Checkpoint/Restart may be more effective
  - How to end test (deal with tail)

Other Steps

- Add explicit I/O Load to test
- Simulate more accurate user behavior
  - Submit a job that aborts
  - Submit jobs with inaccurate ROM times
- Run test with current scheduling methods on T3E
- Determine how to accurately incorporate interactive load
- Run baseline on NERSC-3
- Determine proper stopping criteria
  - Minimize end of test idle (real systems always have more work)
- Move to a generalized framework
  - Evaluate more systems

THE ESP FRAMEWORK

Going to General Framework

- Fine tune ESP implementation
- Incorporate the ability to use other workload applications
- Investigate feasibility of adding more “common” tests
- Run test on more architectures
- Determine ways to normalize results
  - Different sizes of systems
  - Different system types
  - Different workloads/applications
- Improve test to prevent “gaming”
- Write specific rules
  - Framework may be a “pen and paper test”
    - See if there can be a reference implementation

Ways to Improve Effectiveness

- Decrease overhead in starting jobs
  - MPI/Batch Launch times
- Improve functionality
  - System Wide Checkpoint Restart
  - Improve scheduling
  - Coordinated priorities
- Improve shutdown and start up times
- Improve ability for system upgrades
- Etc.

Summary

- Effective system determination is extremely important in our cost conscious world
  - Both before and after purchase
- NERSC is implementing an ESP test to determine and compare how effective large systems are
- This or similar methods will be used to project effectiveness and thereby influence vendors
- NERSC has discussed this idea with several vendors and all say this is the right thing to do and indicate willingness to consider participating
  - IBM committed to specific goals for NERSC-3

Conclusions

Conclusions

- To be effective, a 21st Century facility must have capability systems and be an intellectual leader for large scale science
- Over the past several years, the fundamental methods of modeling have been proven to run in parallel effectively on hundreds of processors
- With the correct architectures, very high quality service and efficiency can be delivered in a highly parallel environment
- Running the systems and facilities of the future with the methods of the past will not work
- Change will occur with increasing frequency so a site must create the infrastructure for easily accommodating change
- A facility must use its excellent staff and influence to expand its role with new initiatives and expansion of projects